LDA & QDA

1. 数统角度剖析（贝叶斯）
2. 一维的 LDA
3. 多维的 LDA
4. QDA
5. 非数学角度解释
6. Logistics Regression, LDA, QDA 对比
7. 分类器评估

LDA / QDA 详细整理

☆ 数统角度剖析

记得 Logistic Regression 用 sigmoid function 直接估计 $P(Y=k|X=x;\theta)=\frac{1}{1+e^{-\theta x}}$

现在考虑一种间接的方式来估计。

记得贝叶斯公式，

$$P(A_i|B)=\frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)}$$

其中 $P(A_i)$ 是先验概率，因为它不需考虑B方面的因素.

在这里，$P(A_i) \Rightarrow P(Y=k) \Leftrightarrow \pi_k$

π_k 代表一个随机抽样是来自第K类的先验概率

（一共有10000个样本，6000个属于A类，3000个B类，1000个C类，

那么 $\pi_A=\frac{6000}{10000}=0.6$，$\pi_B=0.3$，$\pi_C=0.1$）

而 $P(B|A_i) \Rightarrow P(X=x|Y=k) \Leftrightarrow f_k(x)$

$f_k(x)$ 表示在对应的每个类别的情况下，相应的X的分布

（已知选定A类，$P(X|Y=A)$ 代表6000个sample的分布）

$\Leftrightarrow$ the density function of X for an observation that comes from the kth class.

$\Leftrightarrow$ $f_k(x)$ is relatively large if there is a high probability that an observation in the kth class has $X \approx x$

综上：

$$P(Y=k|X=x)=\frac{\pi_k f_k(x)}{\sum_{l=1}^{k}\pi_l f_l(x)}$$

估计或计算 π_k 很简单，而估计 $f_k(x)$ 很难. 除非我们用一些简单的分布代替.

☆首先, LDA for $P=1$ (we only have 1 predictor)

假设 $f_K(x)$ 是 normal distribution.

$$f_K(x) = \frac{1}{\sqrt{2\pi}\sigma_K} \exp\left(-\frac{1}{2\sigma_K^2}(x-\mu_K)^2\right)$$

where μ_K, σ_K^2 are the mean and variance for Kth class.

这里, 假设 $\sigma_1^2 = \cdots = \sigma_K^2$

那么

$$P_K(x) = P(Y=k \mid X=x) = \frac{\pi_K \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2\sigma^2}(x-\mu_K)^2\right)}{\sum_{l=1}^{K}\pi_l \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)} \quad *$$

The Bayes classifier involves assigning an observation $X=x$ to the class for which * is largest.

在对 * 作对数变换和整理后:

$$\delta_K(x) = x \cdot \frac{\mu_K}{\sigma^2} - \frac{\mu_K^2}{2\sigma^2} + \log(\pi_K)$$

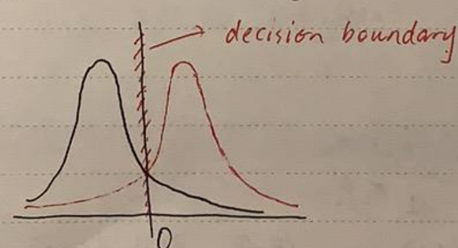
举例: if $K=2$, $\pi_1 = \pi_2$

$$\Rightarrow \quad \delta_1(x) = x \cdot \frac{\mu_1}{\sigma^2} - \frac{\mu_1^2}{2\sigma^2} + \log(\pi_1)$$

$$\delta_2(x) = x \cdot \frac{\mu_2}{\sigma^2} - \frac{\mu_2^2}{2\sigma^2} + \log(\pi_2)$$

如果 $\delta_1(x) > \delta_2(x)$, 归为 class 1. $\Leftrightarrow$ $2x(\mu_1 - \mu_2) > \mu_1^2 - \mu_2^2$

那么 Decision Boundary $\Rightarrow$ $x = \frac{\mu_1^2 - \mu_2^2}{2(\mu_1 - \mu_2)} = \frac{\mu_1 + \mu_2}{2}$



但真实情况下, $f_K(x)$ 不一定是 Gaussian, 就算是 Gaussian, $\mu_1, \cdots, \mu_K$, $\sigma_1^2, \cdots, \sigma_K^2$ (这里都是 σ^2), $\pi_1, \cdots, \pi_K$, 都要估计.

但是 LDA 对 Bayes Classifier 通过以下方式来估计.

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

其中, n_k 是在k类中的所有样本数, $\hat{\mu}_k$ 就是k类中样本均值
$\hat{\sigma}^2$ 看作是每个类的样本方差的加权平均.

而 $\hat{\pi}_k = n_k/n$. (n 为样本总数)

把这些 estimates 带到 $\delta_k(x)$ 中

$$\Rightarrow \hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\mu_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

对于一个 observation, ~~~~ 把它归到使 $\hat{\delta}_k(x)$ 最大的那类。

LDA 中的 L ⇒ linear 意思是 $\delta_k(x)$ 是 x 的线性方程.

☆ 接下来, p>1 的时候呢?

⇒ $X = (X_1, X_2, \cdots, X_p)$ is drawn from a multi-variate ~~~~ Gaussian Distribution, with a class-specific mean vector and a common cov matrix.

Multivariate Gaussian 假设每一个 predictor (X_i) 都服从一维的 Gaussian. 其中每对 predictor ($\{X_i, X_j\}$) 有相关性.

To indicate that a p-dimensional random variable X has a multivariate Gaussian distribution, we write $X \sim N(\mu, \Sigma)$ $\mu = E(X)$, $\Sigma = Cov(X)$

$$f(x) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

同理作一些处理之后

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$

使得，对于一个给定的 observation，我们把它归到使 $\delta_k(x)$ 最大的类

示意图如下

dash lines 代表 decision boundary，或者说

它们代表满足 $\delta_k(x) = \delta_l(x)$ $(k \neq l)$

同样我们也要估计 $\mu_1, \cdots, \mu_K, \pi_1, \cdots, \pi_K, \Sigma$。

对于实际应用来说，LDA 所得到的分类结果，往往~~会~~有 lowest total error out of all classes ⇒ Sensitivity / Specificity 会很低

这种情况下，要适应调整 threshold，来提高对某个类别的分类准确度。（怎么选 threshold？ Grid Search）

BONUS: How to evaluate a classification model performance?

① ROC 是一种检验方法 ⇒ ROC traces out two types of error as we vary the threshold value.

② confusion matrix ⇒ 提炼多个指标（sensitivity, specificity, recall, accuracy ...）

③ 调整 threshold，上面说过

④ AUC，其实就是 ROC 曲线下面的面积。

☆ QDA ⇒ Quadratic Discriminant Analysis

比较： LDA QDA $(p>1)$.

	LDA	QDA	
(1)	$\delta_k(x)$ 是 x 的线性函数	~ 二次函数	
(2)	σ^2 / Σ 是一样的	~ 不一样	一个意思.
(3)	$X \sim N(\mu_k, \Sigma)$	$X \sim N(\mu_k, \Sigma_k)$	
(4)	decision boundary 线性	~ 非线性.	

QDA 中的 $f_k(x)$ 在 log 变换后

$$\Rightarrow \delta_k(x) = -\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k) - \frac{1}{2}\log|\Sigma_k| + \log \pi_k$$

$$= \underbrace{-\frac{1}{2}x^T\Sigma_k^{-1}x}_{\text{Quadratic}} + x^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma_k^{-1} - \frac{1}{2}\log|\Sigma_k| + \log\pi_k$$

??? 所以为什么 $\Sigma \to \Sigma_k$ 是很重要的？

⇒ Bias-variance ~~trad~~ trade-off

在 Σ 不一样时，QDA 在对 Σ 求解时，需很大计算量，所以 LDA 要比 QDA less flexible（甚至它们复杂程度差别较大）

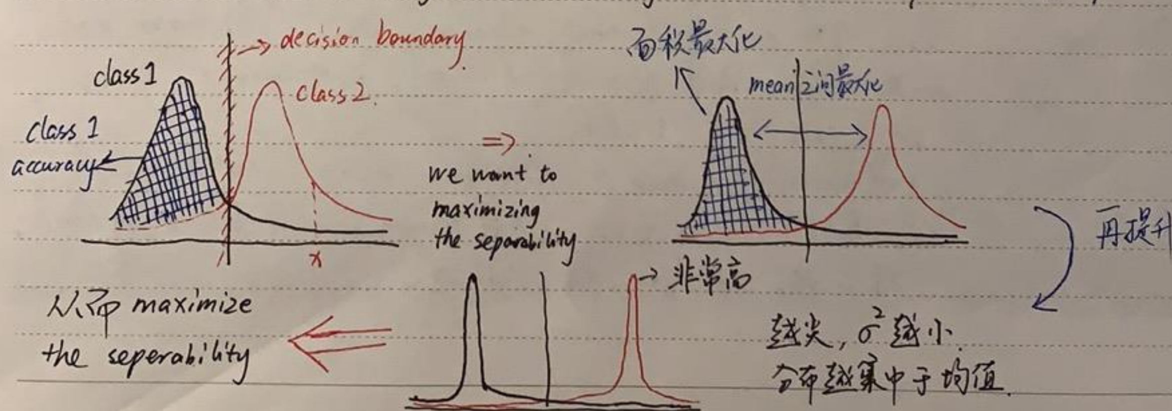
??? 所以，什么时候用 LDA / QDA？

training set is very large → QDA.

〜 relatively small → LDA（避免 overfitting）

☆ 下面在从非数学角度，通俗来讲讲一下 LDA . QDA

(QDA)
LDA ⇒ maximizing the seperability between the two / multi groups



也就是要 max $\frac{(\mu_1-\mu_2)^2}{S_1^2+S_2^2}$ $\rightarrow$ $\frac{\text{ideally large}}{\text{ideally small}}$

这里的 1, 2 代表类别.

(和PCA很像, 目的是 project data on lower dimension to maximize separation)

☆ 最后, 把 logistic Regression, LDA, QDA 进行对比.

Q&A:

Q1: 既然 logistic Regression 不错, why LDA?

A1: 1) 当类别是 well-separated (没有或很少 overlapping), LR 不稳定

2) 当 n 小, $f(X=x)$ 是 normally distributed, LDA 更稳定

3) LR 很难处理 multi-classification.

Q2: logistic Regression 和 LDA 有什么相似么?

A2: 注意: 对于 LR, $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} \Rightarrow \log\left(\frac{P_1}{1-P_1}\right) = \theta^T x$ 可视作为 $(\beta_0+\beta_1 x)$

这里 $P_1 = h_\theta(x) = P(Y=k \mid X=x;\theta)$

$\frac{P_1}{1-P_1} \Rightarrow$ odds

对于 LDA: $\log\left(\frac{P_1}{1-P_1}\right) = C_0 + C_1 X$, 其中 C_0, C_1 都是 μ_1, μ_2, σ^2 的函数 (不推导了)

可看出, LR 和 LDA 的 decision boundary 都是线性的.

只不过 β_0, β_1 通过 MLE 估计的, C_0, C_1 通过估计 μ_1, μ_2, σ^2.

Q3: 它们俩何时使用?

A3: 首先, 可以对每个 feature (numerical) 做一个 distribution plot, 如果 feature 满足 normally distributed, 且 training set 不大, LDA 更好. 否则 LR 更好.

(如果 decision boundary 是 highly non-linear, KNN 是好选择)

QDA 是介于它们之间的, 它的 decision boundary 是 quadratic.

实操的话, 10-fold CV 大法好!

LDA/QDA example

Zijing Gao

4/21/2020

```r
# load the data
library(ISLR)

# train test split
train_idx = sample(nrow(iris), 0.8*nrow(iris))
train = iris[train_idx,]
test = iris[-train_idx,]
```

```r
library(MASS)

lda.fit = lda(Species~., data = train)
```

```r
cbind(prior = lda.fit$prior,
      counts = lda.fit$counts)
```

```
##                prior counts
## setosa     0.3416667     41
## versicolor 0.3166667     38
## virginica  0.3416667     41
```

```r
prop = lda.fit$svd^2/sum(lda.fit$svd^2)
prop
```

```
## [1] 0.990934147 0.009065853
```

We can use the singular values to compute the amount of the between-group variance that is explained by each linear discriminant. In our example we see that the first linear discriminant explains more than 99% of the between-group variance in the iris dataset.

```r
# predict with test data
pred.lda = predict(lda.fit, test[,1:4])
table(pred.lda$class, test$Species)
```

```
##
##              setosa versicolor virginica
##   setosa          9          0         0
##   versicolor      0         12         0
##   virginica       0          0         9
```

Perfect!

```r
# set CV = TRUE
lda.cv = lda(Species~.,data = iris, CV = TRUE)
table(lda.cv$class, iris$Species)
```

```
##
##              setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         1
##   virginica       0          2        49
```

```
cat("accuracy:", sum(diag(table(lda.cv$class, iris$Species))) / sum(table(ld
a.cv$class, iris$Species)))
```

```
## accuracy: 0.98
```

```
qda.fit = qda(Species~., data = train)
pred.qda = predict(qda.fit, test[,1:4])
table(pred.qda$class, test$Species)
```

```
##
##              setosa versicolor virginica
##   setosa          9          0         0
##   versicolor      0         12         0
##   virginica       0          0         9
```

```
qda.cv = qda(Species~.,data = iris, CV = TRUE)
table(qda.cv$class, iris$Species)
```

```
##
##              setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         47         1
##   virginica       0          3        49
```

```
cat("accuracy:", sum(diag(table(qda.cv$class, iris$Species))) / sum(table(qd
a.cv$class, iris$Species)))
```

```
## accuracy: 0.9733333
```